Document Category: BP

Version: 0.5

Status: Draft

Last Updated: Mar. 7,2017

Solace EnterpriseStats

Best Practices Guide

Contents

[Contents 2](#_Toc490573745)

[1 Introduction 4](#_Toc490573746)

[1.1 Scope 4](#_Toc490573747)

[1.2 Related documents 4](#_Toc490573748)

[1.3 Terminology 4](#_Toc490573749)

[2 EnterpriseStats 5](#_Toc490573750)

[2.1 Problem Statement 5](#_Toc490573751)

[2.2 Solution Overview 5](#_Toc490573752)

[2.3 Using Stats Receivers for integration with downstream applications 6](#_Toc490573753)

[2.4 End to End Deployment of EnterpriseStats 7](#_Toc490573754)

[2.5 Ownership 8](#_Toc490573755)

[3 Installation Best Practices 9](#_Toc490573756)

[3.1 Deployment Considerations 9](#_Toc490573757)

[3.1.1 Standalone deployment 9](#_Toc490573758)

[3.1.2 Redundant Deployments 9](#_Toc490573759)

[3.1.3 Global Deployments 10](#_Toc490573760)

[3.2 Pre-install requirements Checklist 10](#_Toc490573761)

[3.2.1 Networking Prerequisites 10](#_Toc490573762)

[3.2.2 Configuration Prerequisites 11](#_Toc490573763)

[3.2.3 Others 12](#_Toc490573764)

[4 Configuration & Tuning 13](#_Toc490573765)

[4.1 Adding New SEMP Pollers 13](#_Toc490573766)

[4.2 Monitoring HA Pairs 14](#_Toc490573767)

[4.2.1 Solace Appliances 14](#_Toc490573768)

[4.2.2 Solace VMRs 15](#_Toc490573769)

[4.3 Grouping Pollers based on functionality 15](#_Toc490573770)

[4.4 Control Polling Intervals 16](#_Toc490573771)

[4.5 Tuning InfluxDB for performance 16](#_Toc490573772)

[4.5.1 Retention policies 16](#_Toc490573773)

[4.5.2 Continuous queries 16](#_Toc490573774)

[4.5.3 Sample Retention Policies and Continuous Queries 17](#_Toc490573775)

[4.6 Tuning the InfluxDB Stats Receiver for performance 18](#_Toc490573776)

[4.6.1 Thread Pool Settings for the Stats Receiver 18](#_Toc490573777)

[4.6.2 Specifying the Stats Messages to be written to InfluxDB 18](#_Toc490573778)

[4.6.3 Specifying the fields to be written per stats message 19](#_Toc490573779)

[5 Troubleshooting 20](#_Toc490573780)

[5.1 Common Errors & Troubleshooting scenarios 20](#_Toc490573781)

[5.2 Systematically troubleshooting Missing Data in Grafana Dashboards 20](#_Toc490573782)

[5.2.1 Check if components are running 20](#_Toc490573783)

[5.2.2 Run the Grafana Dashboard query directly on InfluxDB 21](#_Toc490573784)

[5.2.3 Ensure that the InfluxDB database is not full 21](#_Toc490573785)

[5.2.4 Examine the InfluxDB Stats Receiver Configuration 22](#_Toc490573786)

[5.2.5 Run sdkperf to see if you can subscribe to the stats topics 22](#_Toc490573787)

[5.2.6 Isolate your missing data to a Pump poller 22](#_Toc490573788)

[Approvals 24](#_Toc490573789)

[History 25](#_Toc490573790)

# Introduction

## Scope

The Solace PSG Enterprise Stats Solution can be used for effectively monitoring your Solace Ecosystem, for both real-time monitoring as well as historical analysis.

The purpose of this document is to outline the best practices and guidelines for installation, configuration and operation of an EnterpriseStats deployment. The document will not cover the installation, configuration and operational procedures -references will be made to the EnterpriseStats Runbook document which has these procedures documented.

## Related documents

These documents contain information related to the information in this document.

| Document Number | Document Title | Version (Optional) |
| --- | --- | --- |
| [EnterpriseStatsRB] | Enterprise Stats Runbook | 1.2 or above |
|  |  |  |

Table ‑ Related Documents

## Terminology

| Term / Acronym / Abbreviation | Definition |
| --- | --- |
| VPN | Message-VPN |
| HA | High Availability |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table ‑ Terminology

# EnterpriseStats

## Problem Statement

The Solace Element Management Protocol (SEMP) is an open interface for monitoring and managing Solace routers. Solace currently supports 2 versions of SEMP:

* Legacy SEMP or SEMPv1, which use XML over HTTP
* SEMPv2, a RESTful API for monitoring, managing and configuring Solace

SEMPv2 is being released iteratively and does not support monitoring commands at the time of writing this document. Legacy SEMP fully supports the monitoring, management and configuration of Solace and is used by EnterpriseStats for monitoring.

Using SEMPv1 or Legacy SEMP, monitoring applications can poll Solace routers over management or the message backbone by sending XML requests over HTTP to retrieve rich fine grained router statistics and information. The data being returned from SEMP can be stored, displayed and analyzed for trends. SEMP thus enables a proactive monitoring mechanism and can be used for populating real time dashboards, historical trending and capacity management. For more information on concepts, examples and polling guidelines on legacy SEMP, refer to:

<http://docs.solace.com/SEMP/Using-Legacy-SEMP.htm>

The SEMP schema is distributed with each and every SolOS load and can be used for parsing SEMP responses.

The main issues when monitoring using SEMP are as follows:

* **What to poll:** SEMP is an extremely powerful tool for retrieving statistics from a Solace router – its scope ranges from System wide statistics on router health and hardware status, to detailed per message-vpn and queue statistics. Monitoring applications may not be aware of which SEMP request to send Solace in order to retrieve a certain piece of monitoring information, as this requires an in-depth understanding of the detailed statistics generated by Solace routers; and how to compose this SEMP request from the SEMP schema
* **How to parse:** Due of the SEMP interface’s power and breadth of scope, there is a significant learning overhead to master the parsing of SEMP responses.
* **Polling too frequently, too many times:** Multiple applications may be polling the Solace router for the same information, causing it to hit the threshold for total number of concurrent SEMP requests. As the number of users of SEMP increases, the load on the SEMP engine goes up.
* **Duplication of effort:** Multiple monitoring teams may be duplicating effort building applications for SEMP polling by polling for the same information, and building their own individual SEMP parsers

## Solution Overview

EnterpriseStats is a Solace Professional Services solution to solve the challenges with SEMP-based synchronous monitoring - it combines centralized SEMP-based monitoring with Solace’s efficient asynchronous topic-based message routing for fan out and distribution to subscribers over the message-bus in an easy to understand message-format.

The core of the Solace PSG Enterprise Stats Solution is the “StatsPump”, a SEMP Data republishing tool which can be used for real-time topic based monitoring of the various statistics generated by the Solace router. The StatsPump polls Solace routers for statistics using the Solace “SEMP” protocol on the control plane of the router, reformats the data into one of many configurable formats such as JSON, XML, SDTMap, etc., and republishes these statistics on a defined message bus for ease of consumption on pre-defined topics. Special “Stats Receivers” can subscribe to these messages for monitoring purposes.

Monitoring applications no longer need to poll Solace routers for information, and simply need to subscribe to a topic of interest using the API of their choice to receive statistics in an asynchronous fashion in an easily digestible message format.

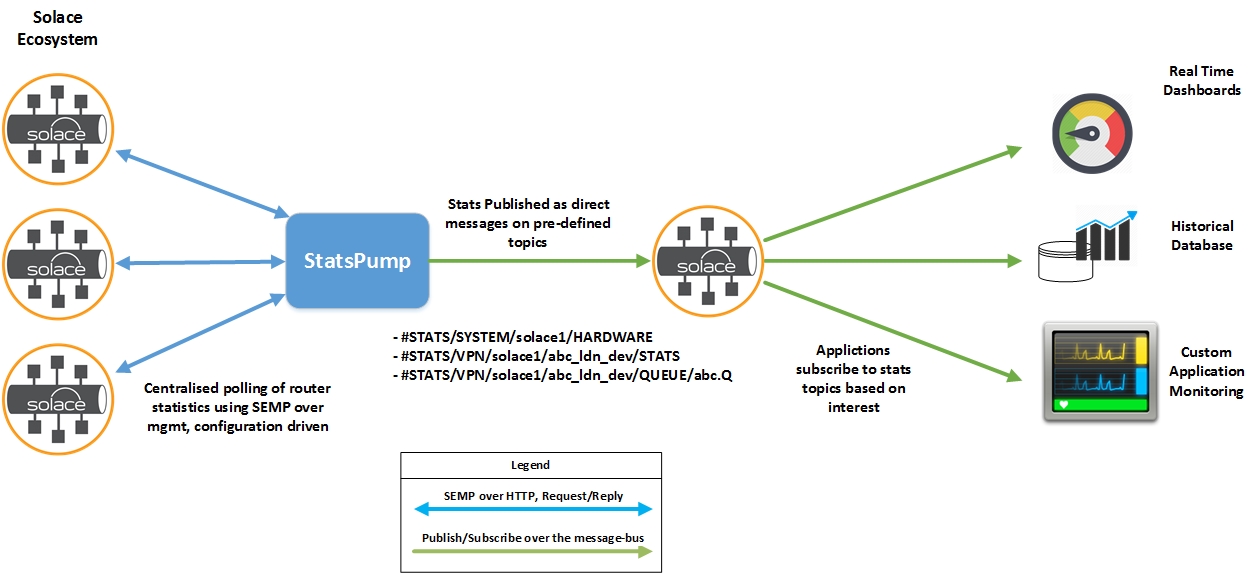


Figure ‑: Enterprise Stats Logical Diagram

Figure 2‑1 illustrates the logical deployment for EnterpriseStats. The Solace Ecosystem to be monitoring is depicted in the left side of the figure. StatsPump polls the Solace routers in a centralized fashion for information using SEMP over HTTP. It then parses and transforms this information to into one of many configurable formats such as JSON, XML, SDTMap, and publishes this to predefined topics over the message-bus. Solace recommends publishing statistics information to a separate Solace router or VMR to ensure that statistics messages do not interfere with the Solace ecosystem being monitored. Monitoring applications connect to the VPN on which EnterpriseStats is publishing statistics and subscribe the topic of their interest.

Monitoring applications are thus no longer required to compose and send SEMP requests or perform the complicated task of parsing SEMP responses or translating them to different formats. This results in an efficient fan out of statistics from a Solace router out to all interested monitoring clients, using the in-built topic routing capabilities of Solace. In addition, the load on the SEMP engine on a single Solace router is reduced and the XML parsing and transformation is offloaded from the monitoring clients.

## Using Stats Receivers for integration with downstream applications

The statistics received from StatsPump can be used for a variety of applications – such as storing in databases for historical trending, populating real-time dashboards, big data systems for analytics, etc. These systems may or may not provide the functionality to connect to the Solace message bus and subscribe to statistics.

Solace Professional Services provide Stats Receivers which allow integration between StatsPump and downstream applications. Stats Receivers subscribe to statistics messages published by StatsPump using Solace developer best practices, transform the message into a format expected by a downstream system and write them to the downstream application. At the time of writing of this document, Stats Receivers are available for storing in various databases such as InfluxDB and Hadoop – for more information on available Stats Receivers refer to the link:

<http://statsdemo.solace.com/plugins/>

Solace Professional Services provide a framework for Stats Receivers which allows you to easily write a new Receiver and integrate it with the StatsPump to receive statistics. For more information, refer to Section 6

## End to End Deployment of EnterpriseStats

In order to build an end-to-end monitoring solution, the statistics generated by StatsPump needs to be stored in a database, and a graphing dashboard is needed for charting this data. InfluxDB (<https://www.influxdata.com>) is a time-series database, built specifically for storing this type of data. Grafana (<http://grafana.org>) is a web-based dashboard for building charts from time-series data. Many Solace customers have integrated Solace statistics with InfluxDB and Grafana, and this ideal pairing of technologies with the EnterpriseStats is used as an example to build an end-to-end monitoring solution of the Solace Ecosystem.

The below figure illustrates the various logical components in the EnterpriseStats monitoring solution, and the flow of data between them.

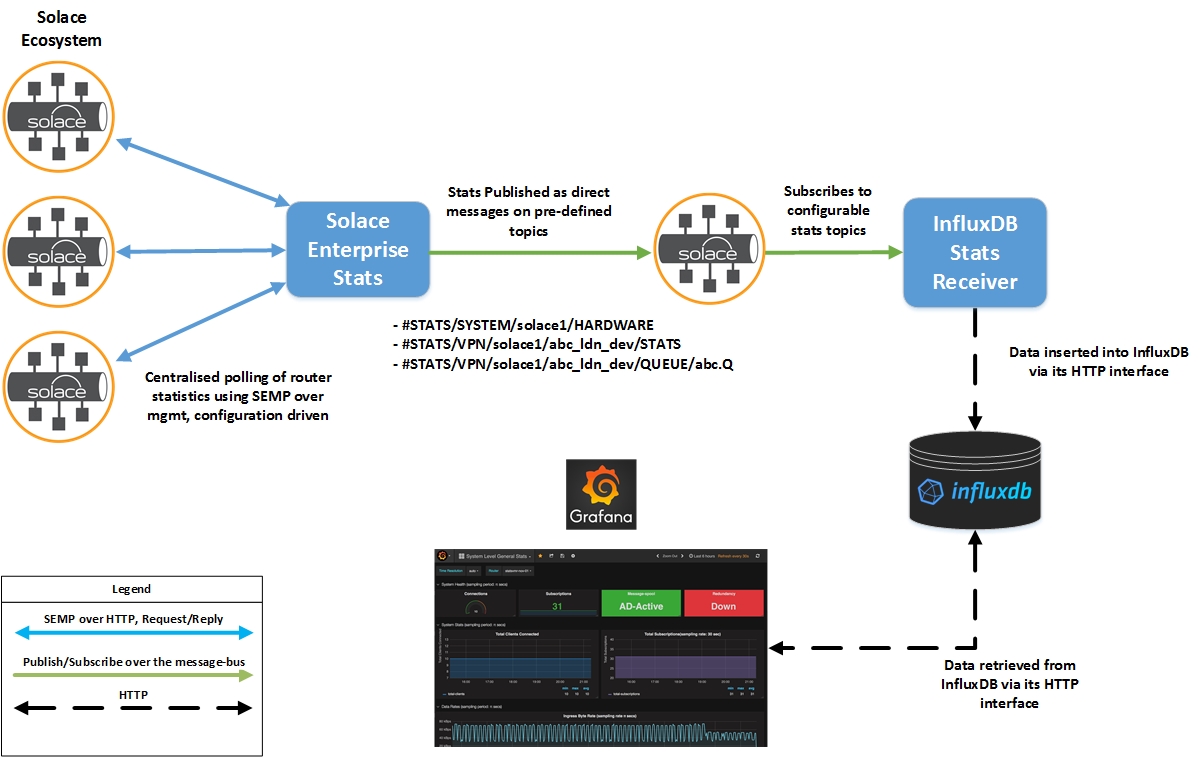


Figure ‑: End to End monitoring using EnterpriseStats

The Solace routers to be monitored by StatsPump are depicted at the left of the figure. Three routers in a standalone mode are shown for the sake of keeping the diagram simple, but these can be extended to more routers, including those deployed as an HA pair.

StatsPump runs as a standalone Java application on a host separate from the Solace router, and should be able to reach the management interface of Solace router(s) being monitored, as well as the message-backbone interface of the Solace router(s) the transformed statistics are re-published to. SEMP over the management interface is used to retrieve statistics from the Solace router(s) being monitored by EnterpriseStats. StatsPump will be configured to republish statistics from all Solace routers to a VPN on a separate Solace router or VMR - this is done in order to isolate the statistical data from the production environment being monitored and prevent the statistical information being published by StatsPump from consuming resources on the routers being monitored or skewing their statistics.

StatsPump polls the Solace routers being monitored using SEMP over HTTP, parses the statistics, transforms them to the desired formats and re-publishes them on pre-defined topic, such as:

* #STATS/SYSTEM/<router-name>/<statistic>
* #STATS/VPN/<router-name>/<vpn-name>/<statistic>
* #STATS/VPN/<router-name>/<vpn-name>/CLIENT/<statistic>
* #STATS/VPN/<router-name>/<vpn-name>/QUEUE/<statistic>

The InfluxDB Stats Receiver is a standalone application that connects to the message-vpn on Solace routers that the StatsPump is publishing statistics to, and subscribes to the topics as desired. It then composes an insert statement using InfluxDB's Line Protocol, and posts this to InfluxDB using its HTTP API.

Finally, dashboards are developed in Grafana to visualize the information as desired. Grafana uses InfluxDB's HTTP API to retrieve this data from the database, and displays this over its web-interface to end-users.

## Ownership

The EnterpriseStats Solution developed by Solace’s Professional Services Group has an associated licensing cost that is incremental to the purchase of Solace routers. Upon purchasing the software, customers will receive a license agreement that outlines the terms and condition specific to IP ownership, warranties, usage, etc…

EnterpriseStats Solution is not managed by the Solace Product management team or supported by the Technical Support team. Customers can purchase additional professional services in order to install and configure the EnterpriseStats solution, create additional Stats Receivers, make feature enhancements, debug any issues, etc.

# Installation Best Practices

The installation procedure for all the components in Enterprise Stats is documented in the [EnterpriseStatsRB] document, Section 4: Installation.

The following sections describe the best practices for installation:

## Deployment Considerations

### Standalone deployment

The components in the EnterpriseStats solution are as follows:

* StatsPump application
* InfluxDB Stats Receiver
* InfluxDB
* Grafana

The first two components run as standalone Java applications, while the third and fourth are products which need to be installed separately and run as a service.

The InfluxDB Stats Receiver and InfluxDB must be deployed on separate hosts on the same LAN, to minimize any latency in the Stats Receiver writing the statistics information to the database. Ideally, Grafana must also be running on the same host as the database, to minimize any lag when statistics information is being retrieved from InfluxDB by Grafana.

It is recommended to deploy the InfluxDB Stats Receiver and the EnterpriseStats application on different hosts on the same LAN, to keep the source of the monitoring information being published separate from the InfluxDB Stats Receiver application; This strategy will also prevent high resource utilization on the host by Grafana or InfluxDB due to increased read/writes to InfluxDB, having an impact on the operation of EnterpriseStats.

When configuring the location which EnterpriseStats publishes statistics, Solace recommends publishing statistics information to a separate Solace message router to ensure that statistics messages do not interfere with the Solace ecosystem being monitored.

For more information, refer to [EnterpriseStatsRB], Sec 2.6.2: Simple Management Stats Bus Configuration.

### Redundant Deployments

StatsPump and Stats Receiver do not provide any built-in redundancy feature, or deployments in a clustered fashion. The applications are stateless – if one instance dies, another one with the same configuration should be brought up and it will continue operation where it left off. This can be achieved easily using existing monitoring applications in the customer’s enterprise such as ITRS or Nagios.

Redundancy can be achieved for the other components in EnterpriseStats as follows:

* The Solace routers or VMRs to which the StatsPump is publishing statistics can be deployed as a redundant pair to achieve high-availability
* InfluxDB and Grafana both provide their own clustering and redundancy solutions. For more information, refer to the below links:

<https://docs.influxdata.com/influxdb/v1.1/high_availability/relay/>

For more information, refer to [EnterpriseStatsRB], Sec 2.6.3: Fully Redundant Enterprise Stats – One region.

### Global Deployments

When monitoring a Solace ecosystem consisting of Solace routers deployed in multiple geographic regions, it is recommended to deploy multiple instances of the EnterpriseStats solution, with one dedicated instance in each geographic region. If deployed only in one geographic region, StatsPump will require SEMP polling over the WAN to reach Solace routers in a remote region. Since the centralized SEMP polling operation is synchronous, this becomes a bottleneck for monitoring global applications and will result in lower polling frequencies. It recommended to minimize SEMP polling over the WAN.

For example, if the Solace ecosystem consists of Solace routers in North America (East) and London, one instance of the EnterpriseStats should be deployed in each region. Hence, each region will have its own instance of:

* StatsPump (for performing SEMP polling within the region),
* InfluxDB Stats Receiver (for subscribing to statistics and inserting them into InfluxDB),
* InfluxDB (for storing historical data for Solace routers in this region) and
* Grafana (for dashboard display of statistics for Solace routers in this region).

This option allows local teams to collect monitoring statistics on a per-region basis.

However, it may be required to provide a global view of statistics. This can be implemented use the multi-node routing feature of Solace to share a common statistics message-bus. Each region will run its own instance of StatsPump that will poll routers in the local region for statistics, as well as a Solace Router/VMR to which statistics are published to. All these Solace routers will be added as MNR neighbors to share the statistics messages, and a single instance of the InfluxDB Stats Receiver will receive global statistics and write them to InfluxDB.

For more information, refer to [EnterpriseStatsRB], Sec 2.6.3: Global Enterprise Stats.

## Pre-install requirements Checklist

### Networking Prerequisites

|  |  |
| --- | --- |
| **EnterpriseStats** | |
| Management Connectivity | The host on which StatsPump is deployed on, should be able to reach the management interface of all Solace routers being monitored, in order to retrieve information from them using SEMP over HTTP |
| Message-Backbone Connectivity | The host on which StatsPump is deployed on, should be able to reach the message-backbone interface of the Solace router to which monitoring statistics are being published to, on pre-defined topics |
| **InfluxDB Stats Receiver** | |
| Message-Backbone Connectivity | The host on which the InfluxDB Stats Receiver is deployed on, should be able to reach the message-backbone interface of the Solace router to which monitoring statistics are being published, to subscribe to statistics |
| InfluxDB HTTP API (default port 8086) | The host on which the InfluxDB Stats Receiver is deployed on, should be able to reach InfluxDB’s HTTP API endpoint in order to write statistics to the database |
| **InfluxDB** | |
| InfluxDB HTTP API (default port 8086) | The InfluxDB Stats Receiver and Grafana should be able to access the port for the InfluxDB HTTP API on the host running InfluxDB to read and write to the database. |
| InfluxDB admin interface (default port 8083) | The host on which InfluxDB is deployed should allow incoming connections on the InfluxDB Admin Interface (default port 8083) to configure InfluxDB |
| **Grafana** | |
| InfluxDB HTTP API (default port 8086) | The host on which Grafana is deployed on, should be able to reach InfluxDB’s HTTP API endpoint in order to read statistics from the database |
| Grafana Admin Interface (default port 3000) | The host on which Grafana is deployed should allow incoming connections on the Grafana Admin Interface (default port 3000) to view and configure dashboards |

Figure ‑: Networking Prerequisites

### Configuration Prerequisites

The following configurations must be created on the Solace Ecosystem being polled:

|  |  |
| --- | --- |
| **Solace Ecosystem being monitored** | |
| Item | Description |
| CLI Username | A read-only management CLI account should be created on each Solace router that StatsPump is supposed to poll. This should have a minimum access-level of read-only |
| **Solace router to which Stats are published** | |
| Item | Description |
| Message-VPN | A message-vpn should be created, to which StatsPump will publish statistics messages |
| Client Username, Client Profile and ACL Profile for EnterpriseStats | A client username must be created on the message-vpn, to allow StatsPump to connect and publish stats messages |
| Client Username, Client Profile and ACL Profile for InfluxDB Stats Receiver | A client username must be created on the message-vpn, to allow the InfluxDB Stats Receiver to connect and subscribe to stats messages. If using guaranteed delivery to receive messages, the client profile must be modified accordingly to guaranteed message reception. |

### Others

* The minimum supported JRE version for EnterpriseStats is JRE v1.7.
* Grafana requires the Initscripts and FontConfig packages in order to run
* Refer to the below link for InfluxDB Sizing. The system requirements depend on the amount of data being stored in the database; it is recommended to deploy InfluxDB on a node in accordance with the guidelines for the Low-Moderate load.

<https://docs.influxdata.com/influxdb/v1.1/guides/hardware_sizing/>

# Configuration & Tuning

The configuration procedure for all the components in Enterprise Stats is documented in the [EnterpriseStatsRB] document, Section 6: Configuration Procedures.

The following sections describe the common configuration tasks and best practices for configuration.

The functionality of StatsPump can be configured through 3 XML configuration files:

* Pollers Configuration
* Poller Groups Configuration
* Router Configuration

The Pollers configuration file (pollers.xml) is used to control SEMP requests that StatsPump uses to retrieve monitoring information from Solace routers, as well as the topics on which this information is published containing details that StatsPump will retrieve from Solace Routers via SEMP. This file contains a number of "Pollers" - each Poller contains details of the SEMP request EnterpriseStats will send to a Solace router, and how the SEMP response is to be parsed.

Related Pollers defined in the pollers.xml file can be classified into poller groups, for the ease of configuration and operation, such as “Hardware Statistics”, “Direct Messaging Pollers” and so on. Routers to be monitored can then be tagged with poller groups, rather than the individual pollers themselves. In addition, groups can be used to control the polling interval of the individual pollers.

The Routers configuration file is used to indicate the details of Solace routers to be polled by StatsPump, such as IP address, CLI account name, as well as details of the Solace router where the statistics information is to be published, such as IP address, message-vpn name, client-username, etc.

## Adding New SEMP Pollers

Each installation of StatsPump comes with a default set of Pollers for receiving common system and message-vpn related statistics on Solace routers; in addition, certain pollers are required for the utility to run:

* show hostname
* show message-spool
* show redundancy detail
* show message-vpn \* stats detail

Certain monitoring applications may require statistics data that is not retrieved by the default pollers. The process of adding new pollers is completely configuration driven and can be easily accomplished by modifying the pollers configuration file.

In order to add a new poller, the following information is required:

|  |  |  |
| --- | --- | --- |
| **Field** | **Value** | **Remarks** |
| Name | Name of the poller, commonly the CLI command equivalent of the SEMP request |  |
| Description | A short description of what the poller does |  |
| SEMP Request | The SEMP request for the poller | In order to retrieve the SEMP request equivalent of a CLI command, use the CLI to SEMP converter, available in each SolOS load |
| Run Configuration | The StatsPump Run configuration for the poller | How do you want the poller to run on an HA pair? Should the poller to run only on the active node in an HA pair at any given time, or on both nodes of an HA pair at all times? |
| Destination | The destination for the statistics messages - Can be one of MGMT, SELF or BOTH to indicate the destination type for publishing the statistics; It should be in accordance with destinations defined in the router configuration | Do you want the statistics information to be published to the MGMT destination (typically for management or middleware information) or do you want this to be published to the SELF destination (typically for app teams) or BOTH? |
| topic-string-suffix | The topic suffix to the main StatsPump topics where the information will be published | Define the suffix to the pre-defined StatsPump topic prefix on which the statistics information will be published, eg “ALARMS” or “STATS” |
| base-tag | The XML path in the SEMP response from which StatsPump should begin parsing of objects |  |

For more information on examples of Pollers, refer to the [EnterpriseStatsRB] document, Section 5.2.1 Pollers Configuration.

## Monitoring HA Pairs

When indicating the details of Solace routers to be monitored by StatsPump in the router configuration file, routers can be specified as:

* Standalone Routers
* HA pairs

Poller run configurations can be used to configure StatsPump to poll only the single active router of an HA pair for certain statistics in order to make the SEMP polling more efficient – this is explained in the following sections:

### Solace Appliances

For Solace routers configured in a High-Availability Pair, each router is polled independently by the StatsPump. StatsPump can be configured to consider a pair in either Active-Standby mode (where no clients using the guaranteed delivery mode connect directly to the backup router), or Active-Active (where both routers are in use and clients can connect to either router – not applicable for Guaranteed Messaging) - this can be configured when specifying router details in the Router Configuration file.

The run configuration assigned to a poller can be used to ensure that StatsPump polls routers in an HA pair efficiently – for example, a poller may be configured to poll both routers in an HA pair independently for all system level statistics such as hardware statistics, but only poll the active router in an active-standby pair for message-spool statistics (since the message-spool is on standby for the mate router). EnterpriseStats uses the output from “show redundancy” to infer which router in an HA pair is active, and applies the poller’s run configuration accordingly.

For more information on run configurations, refer to the [EnterpriseStatsRB] document, Section 5.2.1.1 Run Configuration.

When a failover occurs, the <ROUTER\_NAME> portion of the topic on which statistics are being published will change to the mate name; Therefore, a best practice when designing Stats Receivers is to subscribing to StatsPump messages need to ensure their subscriptions will match both appliance names for the<ROUTER\_NAME> level, for example:

If the individual nodes of an HA pair being monitored are ldnpsol1002n01 and ldnpsol1002n02, the Stats Receiver should subscribe to:

#STATS/VPN/ldnpsol1002\*/STATS

to receive VPN level statistics for both routers.

In an Active-Active configuration, data from both appliances will be broadcast for message-vpn and client statistics, so care must be taken to apply explicit subscriptions for each router to ensure that downstream applications receive information for router that they expect.

### Solace VMRs

At the time of writing this document, Solace VMRs do not support high-availability. StatsPump uses the output of “show redundancy” to determine which is the active router in an HA pair, and since the output of this command when run on a VMR may not be the same as what StatsPump expects, it is unknown whether the StatsPump will respect poller run configurations when polling VMRs.

VMRs can be set up as replication mates in order to simulate HA. In this setup, both the replication mates should be added as standalone routers in the router configuration – StatsPump will poll both routers for statistics information. Client applications should use host-lists with IP addresses for both the VMRs when connecting.

When a DR failover is done from the primary VMR to the replication mate, all client applications will connect to the mate replication site. Since StatsPump is already polling the mate replication site, statistics such as increased incoming connections and subscriptions will be returned by StatsPump received by the Receiver, and in turn written to InfluxDB and be populated on the Grafana dashboards.

## Grouping Pollers based on functionality

The default installation of StatsPump groups pollers based on functionality, such as

* Groups for appliance-related statistics
* Groups for Direct messaging statistics
* Groups for Direct and Guaranteed Messaging statistics
* Separate Groups for Caches and Bridges

Routers can then be tagged with the corresponding group in the router configuration file, without changing the configuration in the Pollers file. Solace recommends maintaining this grouping when adding new pollers.

For more information, refer to the [EnterpriseStatsRB] document, Section 5.2.2 Pollers Groups Configuration.

## Control Polling Intervals

Poller groups allow you to define the interval at which StatsPump will run a poller - adminstrators can use this to control the frequency with which data is retrieved from a router. Solace recommends that high priority statistics such as the number of clients connected to a Solace router, or queue details can be polled with a higher frequency, while lower-priority statistics, such as environment details, can be polled at lower frequencies to keep the overall load on a SEMP engine below its limits.

The default installation of StatsPump contains separate poller groups with lower and higher polling interval of statistics.

For more information, refer to the [EnterpriseStatsRB] document, Section 5.2.2 Pollers Groups Configuration.

## Tuning InfluxDB for performance

As with any database, it is not operationally feasible to store all the data points in InfluxDB for ever, due to storage concerns. It is recommended to retain high resolution raw data points being written to InfluxDB for a certain amount of time, such as the latest day or the latest hour, and down-sample the older data to a lower resolution and store it for a longer time for archival. This will ensure that teach data point being written to InfluxDB is not retained forever, and the database size will not keep growing at a steep rate over time.

This can be implemented using a combination of two features in InfluxDB: Retention Policies and Continuous Queries

### Retention policies

InfluxDB provides a facility whereby the life of the data points in InfluxDB can be configured - once the lifetime of the data point is reached, it is deleted. This can be done by configuring InfluxDB retention policies for a database, depending on how long the data is to be retained. The default retention policy is to store data for ever in InfluxDB.

**NOTE**: Once a retention policy is configured for a database, all data points subsequently written to the database will follow this policy. All data points entered before the retention policy is applied will not respect the new rention policy. It is recommended to configure retention policies for a database prior to starting the InfluxDB Stats Receiver.

### Continuous queries

In addition to retention policies, InfluxDB allows the down sampling of data to a lower precision and storage as a separate table - this can be done by defining Continuous Queries on the data. This feature, when used in conjunction with retention policies, can be used to retain the data in InfluxDB over a period of time. For example, high precision data can be stored for a limited duration such as 30 days, and any data older than 30 days, can be summarized and stored for much longer.

For more information on retention policies and continuous queries, refer to: <https://docs.influxdata.com/influxdb/v1.2/guides/downsampling_and_retention/>

### Sample Retention Policies and Continuous Queries

The retention policies and continuous queries to be configured on InfluxDB would be governed by the business requirements for Data retention and the maximum available size of the database. A sample set of retention policies and continuous queries are indicated below along with the commands to configure them in InfluxDB:

* Store raw data for a day
* Store hourly averages for a week
* Store daily averages for 3 months
* Store weekly averages for a year

#### Creating Retention Policies:

> create retention policy one\_day on statspump duration 1d REPLICATION 1 DEFAULT

> create database statspump\_1h\_avg

> create database statspump\_1d\_avg

> create database statspump\_1w\_avg

> create retention policy one\_week on statspump\_1h\_avg duration 1w REPLICATION 1 DEFAULT

> create retention policy three\_months on statspump\_1d\_avg duration 36w REPLICATION 1 DEFAULT

> create retention policy one\_year on statspump\_1w\_avg duration 52w REPLICATION 1 DEFAULT

#### Creating Continuous Queries:

These are created for System and Message VPN stats – for other statistics, similar commands can be used:

CREATE CONTINUOUS QUERY system\_hourly\_averages ON statspump BEGIN SELECT mean(\*) INTO statspump\_1h\_avg.one\_week.:MEASUREMENT FROM statspump.one\_day./SYSTEM/ GROUP BY time(1h), ROUTER\_NAME END

CREATE CONTINUOUS QUERY vpn\_hourly\_averages ON statspump BEGIN SELECT mean(\*) INTO statspump\_1h\_avg.one\_week.:MEASUREMENT FROM statspump.one\_day./VPN/ GROUP BY time(1h), ROUTER\_NAME,VPN\_NAME END

CREATE CONTINUOUS QUERY system\_daily\_averages ON statspump\_1h\_avg BEGIN SELECT mean(\*) INTO statspump\_1d\_avg.three\_months.:MEASUREMENT FROM statspump\_1h\_avg.one\_week./SYSTEM/ GROUP BY time(1d), ROUTER\_NAME END

CREATE CONTINUOUS QUERY vpn\_daily\_averages ON statspump\_1h\_avg BEGIN SELECT mean(\*) INTO statspump\_1d\_avg.three\_months.:MEASUREMENT FROM statspump\_1h\_avg.one\_week./VPN/ GROUP BY time(1d), ROUTER\_NAME,VPN\_NAME END

CREATE CONTINUOUS QUERY system\_weekly\_averages ON statspump\_1d\_avg BEGIN SELECT mean(\*) INTO statspump\_1w\_avg.one\_year.:MEASUREMENT FROM statspump\_1d\_avg.three\_months./SYSTEM/ GROUP BY time(1w), ROUTER\_NAME END

CREATE CONTINUOUS QUERY vpn\_weekly\_averages ON statspump\_1d\_avg BEGIN SELECT mean(\*) INTO statspump\_1w\_avg.one\_year.:MEASUREMENT FROM statspump\_1d\_avg.three\_months./SYSTEM/ GROUP BY time(1w), ROUTER\_NAME END

## Tuning the InfluxDB Stats Receiver for performance

### Thread Pool Settings for the Stats Receiver

InfluxDB writes data to disk in its write operations, and not to memory. Hence the write performance of InfluxDB is Disk I/O bound. It has been noticed during QA that if the InfluxDB Stats Receiver is configured to subscribe to a large dataset of information such as 10,000 client connections and queues and if the Disk I/O limit is reached, there may be a time lag in Influx writing information to its database as opposed to the actual time of polling of these statistics. When writing large datasets to InfluxDB, it must be sized appropriately or deployed in a cluster.

There are certain parameters in the InfluxDB Stats Receiver configuration which can be used for tuning the performance of the receiver by controlling the number of threads and thread pool size:

* THREAD\_AWAITING\_QUEUE\_SIZE
* THREAD\_POOL\_MINIMUM\_SIZE
* THREAD\_POOL\_MAXIMUM\_SIZE

For more information on the description of each field and sample values, refer to the [EnterpriseStatsRB] document, Section 5.3 Stats Receiver, Table 10 Performance Tuning Options.

### Specifying the Stats Messages to be written to InfluxDB

The “SOLACE\_TOPICS” field in the Stats Receiver configuration file is used to specify the statistics messages that the Stats Receiver gathers and writes to InfluxDB. In order to filter out unnecessary statistic messages, this should include only the topics containing monitoring statistics that downstream applications are interested in.

In its default configuration, the InfluxDB Stats Receiver uses direct messaging to receive statistics messages. If the customer use case requires guaranteed messaging, this can be specified using the SOLACE\_QUEUE parameter in the Stats Receiver configuration file. The topics that the Stats Receiver receives statistics on, should be specified as subscriptions of the queue that the Stats Receiver binds to.

For more information on filtering out unnecessary statistics messages, refer to the [EnterpriseStatsRB] document, Section 5.3 Stats Receiver, Table 9: Stats Receiver Configuration options.

### Specifying the fields to be written per stats message

The InfluxDB Stats Receiver, by default, subscribes to all the fields for a particular type of statistics message. Certain use cases may require the Stats Receiver to only subscribe to a particular field in the SEMP response – such as the number of messages spooled for a queue. The fields that the Stats Receiver should subscribe to can be specified in the filed “DB\_FIELD\_SUBSCRIPTIONS” in the Stats Receiver configuration file. For more information, refer to the [EnterpriseStatsRB] document, Section 5.3 Stats Receiver, Table 9: Stats Receiver Configuration options.

# Troubleshooting

## Common Errors & Troubleshooting scenarios

Common errors encountered when starting up the various components in the EnterpriseStats solution are explained in the EnterpriseStats Runbook. This includes errors such as:

* Connectivity Issues for the StatsPump and InfluxDB Stats Receiver to the Solace management interface or message-bus
* Issues experienced by the InfluxDB Stats Receiver writing to InfluxDB

For more information on examples of Pollers, refer to the [EnterpriseStatsRB] document, Section 9 Troubleshooting Common Issues.

## Systematically troubleshooting Missing Data in Grafana Dashboards

In the EnterpriseStats solution, the statistics of each of the Solace routers being monitored will be used to populate dashboards in Grafana.

A common operational issue is that the expected monitoring data may not be displayed on Grafana (with no errors in dashboard population or component startup) and this may be encountered in one of the below scenarios:

* Configuration changes to StatsPump such as adding new pollers or router configurations
* Changing StatsPump resources such as migration to a newer SEMP version
* Building new dashboards

Errors when implementing changes in the above scenarios may cause the Grafana dashboards not to be populated with the expected data.

There are 4 paths of communication for the flow of data between the components in the EnterpriseStats solution. Expected data would not be populated on the Grafana dashboards if they were lost at one of the message flows between the components. We can debug these components one at a time, from Grafana all the way back to the source to quickly identify which is the one resulting in data not being displayed at Grafana. The suggested sequence of troubleshooting is described in the following sections:

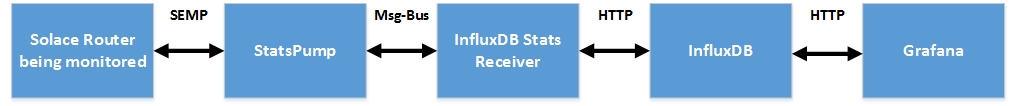


Figure ‑: Debugging EnterpriseStats

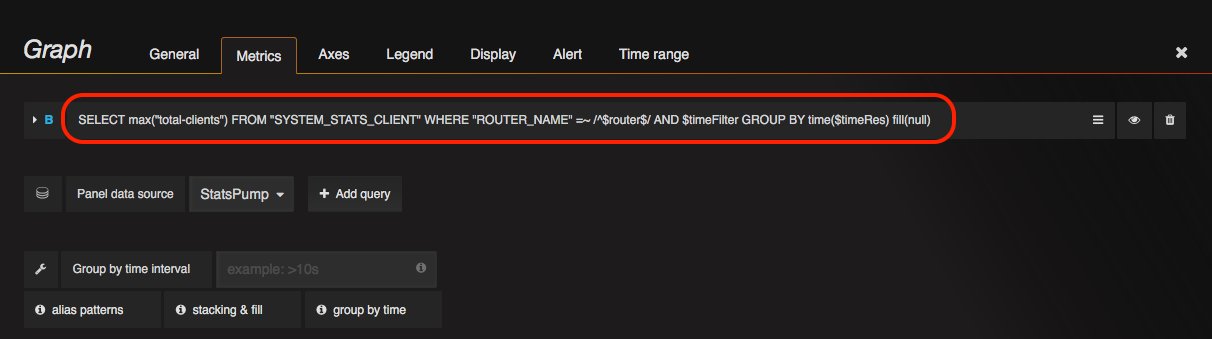
### Check if components are running

The first step is to check if all the components in the EnterpriseStats solution are running, as one of them being shutdown can prevent the flow of data:

1. The Solace routers being monitored must be online and their SEMP-over-management service must be running
2. The StatsPump and InfluxDB Stats Receiver applications must be running
3. InfluxDB and Grafana services must be running

### Run the Grafana Dashboard query directly on InfluxDB

Once it has been established that all components are running, you can select the Grafana dashboard which not displaying the expected data to identify the query used by Grafana to retrieve data from Influx – this can be obtained from the “Metrics” tab of the Dashboard:



Copy this query text and run it directly in InfluxDB, replacing any template variables with their actual values in the query, and removing the “Group by time clause”. Instead, modify the query to return expected values in some time period, such as:

select “total-clients” FROM SYSTEM\_STATS\_CLIENT WHERE “ROUTER\_NAME” = ‘ldnpsol1001n01’ AND time > now()-10m

The above query will retrieve value of the “total-clients” from the SYSTEM\_STATS\_CLIENT series for entries written in the last 10 minutes.

If there is no output returned when running the query directly in InfluxDB, it indicates that the statistics are not even being written to InfluxDB. In this case, or if there is some data returned when running the query but not the expected output, go to the next step.

### Ensure that the InfluxDB database is not full

If the InfluxDB database has grown over time and reached its maximum size, it may explain why no more data can be written to the InfluxDB database. This condition should be accompanied by corresponding errors in the InfluxDB Stats Receiver logs. If the database is full, increase the underlying disk partition size for the database to allow for writing of information to it.

### Examine the InfluxDB Stats Receiver Configuration

If no data is returned when running the dashboard query directly on InfluxDB, check if the InfluxDB Stats Receiver is listening to the correct statistics topics on which StatsPump would publish Statistics messages. This is configured using the “SOLACE\_TOPICS” field:

SOLACE\_TOPICS=#STATS/SYSTEM/\*/>

For more information on the topic hierarchy followed by the StatsPump, refer to the [EnterpriseStatsRB] document, Section 5.1.1 StatsPump publish topic formats.

If some data is returned when running the dashboard query directly on InfluxDB, but is not in the correct format then you can visualize the data textually, by turning back on “INFLUX\_STUBBED” – this will cause the HTTP request to InfluxDB to be written out to a local file instead of being written to the database. The stub file will be located in the root directory of the application, with the filename “InfluxHttp<timestamp>.log”

INFLUX\_STUBBED = true

If the InfluxDB Stats Receiver is subscribed to the correct statistics topics, then proceed to the next step.

### Run sdkperf to see if you can subscribe to the stats topics

If the InfluxDB Stats Receiver component is running properly, the next step is to check if the StatsPump is publishing statistics messages on the topics that the InfluxDB Stats Receiver is subscribed to. You can do this by running a simple sdkperf subscriber client and verifying that data is being received:

./sdkperf\_java.sh –cip=192.168.40.13 –cu=stats@stats –cp=stats –stl=’#STATS/SYSTEM/ldnpsol1001n01/STATS\_CLIENT’ -md

If some statistics data is received but is not what is expected, verify that you are subscribed to the correct statistics topic - refer to the [EnterpriseStatsRB] document, Section 5.1.1 StatsPump publish topic formats.

If statistics data is not received on the statistics topic, it implies that the statistics are not being published by StatsPump – refer to the next step.

### Isolate your missing data to a Pump poller

Lastly, identify the poller corresponding to the statistic on which messages are not being received. Examine the StatsPump poller configuration file if the poller has the correct configuration. Refer to the [EnterpriseStatsRB] document, Section 5.2.1 Pollers Configuration for more information.

Alternatively, if there are issues with StatsPump loading or running the poller, these will appear in the StatsPump application logs. Refer to the [EnterpriseStatsRB] document, Section 9.1.3 Incorrect Configuration in the EnterpriseStats Runbook for more information on identifying these error messages in the logs.

Update the poller configuration if it has errors and restart the StatsPump.

If all the components in the EnterpriseStats solution are found to be working without any configuration errors in the above steps, contact your Solace Account Manager to laisse with Solace Professional Services to fix the issue.

Approvals

This document has been read and approved by the following people, responsible for its implementation. Approval is indicated by an email showing approval. Those approving below indicate that the contents of this document are correct and complete and agree to their implementation:

| Title | Name | Approval |
| --- | --- | --- |
| VP, Global Professional Services | Alex Orsini | Approved |

History

| Version | Status | Date | Author | Reason for changes |
| --- | --- | --- | --- | --- |
| 0.1 | Draft | Dec. 12, 2016 | Shrikanth Rajgopalan | Initial Draft |
| 0.2 | Draft | Jan. 05, 2017 | Shrikanth Rajgopalan | Added Section on Installation Best Practices |
| 0.3 | Draft | Jan. 15, 2017 | Shrikanth Rajgopalan | Added Section on Configuration Best Practices |
| 0.4 | Draft | Jan. 27, 2017 | Shrikanth Rajgopalan | Added Section on Troubleshooting Best Practices |
| 0.5 | Draft | Mar 07,2017 | Shrikanth Rajgopalan | Updates based on PSG Peer review |